

3. TTY Nonvolatile Setups

The RVP8 provides an interactive setup menu that can be accessed either from a serial TTY, or from the host computer interface. Most of the RVP8's operating parameters can be viewed and modified with this menu, and the settings can be saved in non-volatile RAM so that they take effect immediately on start-up. This permits custom trigger patterns, pulsewidth control, matched FIR filter specs, PRF, etc., to be configured by the user in the field.

The TTY menu also gives access to a collection of graphical setup and monitoring procedures that use an ordinary oscilloscope as a synthesized visual display. The burst pulse and receiver waveforms can be examined in detail (both in the time and frequency domain) and the digital FIR filter can be designed interactively to match the characteristics of the transmitted pulse.

3.1 Overview of Setup Procedures

This section describes basic operations within the setup menus such as making TTY connections, entering and exiting the menus, and saving and restoring the configurations.

The setup TTY menu can be accessed by executing the following command:

```
$dspx
```

You will then be prompted by the following:

```
$dspx
Digital Signal Processor 'Chat'
Checking for code upgrades...Okay
(Type ^C to exit Chat Mode)
```

The interactive setup menu is invoked by pressing the **Escape key** on the TTY. If that key can not be found on the keyboard, you can sometimes use Control "[" to generate the ESC code. The RVP8 then responds with the following banner and command prompt.

```
          SIGMET Incorporated, USA
RVP8 Digital IF Signal Processor V3.9(Pol) IRIS 8.03.6
-----
RVP8>
```

The banner identifies the RVP8 product, and displays the RVP8 software version (e.g., V3.9) and IRIS software version (e.g., IRIS 8.03.6). This information is important whenever RVP8 support is required, and it is also repeated in the printout of the "V" command (See below).

The "Q" command is used to exit from the menus and to reload the RVP8 with the (possibly changed) set of current values. It is important to quit from the menus before attempting to resume normal RVP8 operation. Portions of the RVP8 command interpreter remain running while the menus are active (so that the TTYOP command works properly), but the processor as a whole will not function until the menus are exited.

From the command prompt, typing "help" or "?" gives the following list of available commands.

Command List:

- F:** Use Factory Defaults
- S:** Save Current Settings
- R:** Restore Saved Settings
- M:** Modify/View Current Settings
 - Mb** - Burst Pulse and AFC
 - Mc** - Overall Configuration
 - Mf** - Clutter Filters
 - Mp** - Processing Options
 - Mt<n>** - Trigger/Timing <for PW n>
 - Mz** - Transmissions and Modulations
 - M+** - Debug Options
- P:** Plot with Oscilloscope
 - Pb** - Burst Pulse Timing
 - Ps** - Burst Spectra and AFC
 - Pr** - Receiver Waveforms
 - P+** - Visual Test Pattern
- V:** View Card and System Status
- ?:** Print all Menu Commands (this list)
 - ??** - Print all Current Setup Settings
- ***: Sample Receiver Noise Levels
- @:** Display/Change the Current Major Mode
- ~:** Swap Burst/IF Inputs on IFD
- Q:** Quit

3.1.1 Factory, Saved, and Current Settings

The *current settings* are the collection of setup values with which the RVP8 is presently operating; the *saved settings* are the collection of values stored in non-volatile RAM. The saved settings are restored (made current) each time the RVP8 starts. The “**S**” command saves the current settings into the non-volatile RAM, and the “**R**” command restores those non-volatile values so that they become the current settings. The “**F**” command initializes the current settings with factory default values. Thus, “**F**” followed by “**S**” saves factory defaults in non-volatile RAM, so that the RVP8 powers up in its original configuration as shipped.

The RVP8 retains all of its saved settings when new software releases are installed; the new version of code will automatically use all of the previous saved values. However, if the RVP8 detects that the new release requires a setup parameter that did not exist in the previous release, then a factory default value will automatically be filled in for that parameter. A warning is printed whenever this occurs (See also, Section 3.1.2).

There is also support for intermediate minor releases of RVP8 code. Each software release has a major version number (the one that it always had), plus a minor version number for intermediate “unofficial” releases. The minor number starts from zero at the time of each “official” release, and then increments until the next “official” release. The RVP8 includes the minor release number (if it is not zero) in the printout of the “**V**” command. Likewise, the minor release

number of the code that was last saved in the nonvolatile RAM is also shown. This is an improvement over having to check the date of the code to determine which minor release was running.

Note that the RVP8 does not actually begin using the current settings until after the “Q” command is entered, so that the processor exits the TTY setup mode and returns to normal operation.

3.1.2 V — View Card and System Status

The “V” command displays internal diagnostics. This information is for inspection only, and can not be changed from the TTY. The view listing begins with the banner:

Configuration and Internal Status

and then prints the following lines:

RVP8 Digital IF Signal Processor V3.10(Pol) IRIS-8.04

This line shows the revision level of the RVP8 software, the IRIS version.

Settings were last saved using V3.10

This line tells which version of RVP8 code was the last to write into the non-volatile RAM. It is printed only if that last version was different from the version that is currently running. The information is included so that a “smart upgrade” can often be done, i.e., values that did not exist in the prior release can be filled in with a guess that is better than merely taking the factory default.

RVP8 started at: 13:07:33 3 NOV 2003

Current time is: 13:14:03 3 NOV 2003

These lines provide information about when the RVP8 was started, the current system time, and implicitly, the uptime.

CPU-Type: Pentium(R) III

This line displays processor information.

IPP-Library: ippsa6 v2.0 gold SP1 2.0.6.39

This line displays information about the Intel libraries used for RVP8 processing.

Physical hardware inventory:

Found PCI Card RVP8/Rx - Rev.A Serial:1628 Code:14
(/dev/rda/rvp8rx-1)

Found PCI Card RVP8/Tx - Rev.B Serial:1887 Code:9
(/dev/rda/rvp8tx-0)

Found PCI Card I/O-62 - Rev.B Serial:1841 Code:19
(/dev/rda/io62-0)

\--> IO62CP Backpanel - Rev.B Serial:1822 Code:3

The physical hardware inventory provides information about the system hardware being used by the RVP8. This list ONLY displays hardware that is being used, not all hardware in the system (i.e., an RCP8 could be present in the same chasis, but the RCP8 hardware would not be included in this list).

Diagnostics: PASS

If errors were detected by the startup diagnostics then an error bitmask will be shown on the first line. The word "PASS" indicates that no errors were detected.

Processes and Threads:

RVP8Proc-0 - PID:28503 Priority:10 Policy:RealTimeRR

The "Process and Threads" list displays RVP8 processes and their related priority. All RVP8 processes/threads should be running under RealTimeRR policy to guarantee adequate attention from the processors.

Shared library build dates:

This section provides RVP8 developers with information about code resources.

Front panel display:

```
+-----+
| 0.00  AZ/EL  0.00 |
| FFT 100B   1000Hz x1 |
+-----+
```

The front panel display mirrors the display on the front of the RVP8 chassis. This is helpful if you are at a remote location using DspExport.

Tx/Clk:Okay TrigRAM is 99.0% free, TrigCount:378921

The Tx/Clk field displays information about the RVP8/Tx clock (if applicable). TrigRAM provides resource information for those who are implementing custom waveforms.

IFD:Okay Link: Delay = 0.541 usec, Jitter = 0.014 usec

This first section of this line summarizes the receiver status and Burst input signal parameters. The status may show:

Okay RVP8/IFD and connecting cables are all working properly
DnErr Problem in DownLink connection from RVP8/IFD —> RVP8
UpErr Problem in UpLink connection from RVP8 —> RVP8/IFD
NoPLL RVP8/IFD PLL is not locked to external user-supplied clock reference
DiagSW RVP8/IFD test switches are not in their normal operating position

The section second describes the IFD link status. During startup the RVP8 measures the round trip delay along 1) the uplink to the receiver module, 2) the pipeline delays within the receiver module, 3) the downlink, and 4) pipeline delays in the data decoding hardware. The time shown is accurate to within 14ns, and is used internally to insure that the absolute calibration of trigger and burst pulse timing remains unaffected by the distance between the main card and the receiver module. You may freely splice any lengths of cable without affecting the calibrations; the delay time will change, but the trigger and burst calibrations will remain constant.

The standard deviation of the measured delay is also shown. If the link to the IFD is working properly this variation should be less than half the period of its acquisition clock. Larger errors may indicate a problem in the cabling. A diagnostic error bit is set if the error is greater than two acquisition clock periods.

AFC:0.00% (Disabled), Burst Pwr:-48.6 dBm, Freq:30.000 MHz

AFC indicates the level and status of the AFC voltage at the RVP8/IFD module. The number is the present output level in D-Units ranging from -100 to +100. The shorter “%” symbol is used since percentage units correspond in a natural way to the D-Units.

Burst Pwr indicates the mean power within the full window of burst samples. DC offsets in the A/D converter do not affect the computation of the power, i.e., the value shown truly represents the waveform's (Signal+Noise) energy. Freq indicates the mean frequency of the burst, derived from a 4th order correlation model.

For more information refer to Chapter 4.

3.2 View/Modify Dialogs

The **M** menu may be used to view, and optionally to modify, all of the current settings. The current value of each parameter is printed on the screen, and the TTY pauses for input at the end of the line. Pressing Return advances to the next parameter, leaving the present one unchanged. You may also type **U** to move back up in the list, and **Q** to exit from the list at any time.

Typing a numeric or YES/NO response (as appropriate to the parameter) changes the parameter's value, and displays the line again with the new value. All numbers are entered in base ten, and may include a decimal point and minus sign. In some cases, several parameters are displayed on one line, in which case, as many parameters are changed as there are new values entered. In all cases, the numbers are checked to be within reasonable bounds, and an error message (listing those bounds) is printed if the limits are exceeded. Note that changes to the settings (generally) do not take effect until after the **Q** command is typed, at which point the RVP8 exits the local TTY menu and resumes its normal processing operations.

The **M** menu provides access to a large number of configuration settings. As a result, all of the **M** menu commands begin with the letter "M" and are followed by a lower case letter which represents a subcategory, i.e., **Mb** (Burst Pulse and AFC), **Mc** (Overall Configuration), **Mf** (Clutter Filters), **Mp** (Processing Options), **Mt** (Triggers and Timing), **Mz** (Transmissions and Modulations), **M+** (Debug Options). The **??** command by itself prints the entire set of questions so that you can make a hard copy.

The **M** menu always works from the current parameter values, not from the saved values in non-volatile RAM. If the host computer has modified some of the current values, then you will see these changes as you skip through the setup list. However, typing **S** at that point would save all of the current settings and would, perhaps, make many changes to the original non-volatile settings. In general, to make an incremental change to the saved settings, first type **R** to restore all of the saved values, then use the **M** menu to make the changes starting from that point, and **S** to save the new values.

A listing of the parameters that can be viewed and modified with the **M** menu is detailed in the following subsections. In each case, the line of text is shown exactly as it appears on the TTY with the factory default settings. A definition of each parameter is given and, if applicable, the lower and upper numeric bounds are shown.

3.2.1 Mc — Top Level Configuration

This set of commands configure general properties of the RVP8/IFD and RVP8 cards.

Acquisition clock: 35.9751 MHz

This is the frequency of the acquisition sampling clock in the IFD module. This will generally be in the 72MHz range for the RVP8 CAT-5E IFD, and in the 36MHz range for the legacy RVP7 IFD. If you are locking the IFD to an external reference, the center frequency of the installed VCXO is entered (see section 2.2.12).

Limits: 33.33 to 80 MHz

Dual simultaneous receivers are being used: NO

Each receiver is wired to a dedicated IFD: NO

Co-polarized signal is always on the primary Rx: NO

Answer this question “Yes” if the RVP8 will be processing simultaneous signals from two separate receivers. Answering “No” will revert to normal operation with just a single receiver. If dual receivers are being used, then more settings must be configured including if each receiver is wired to a dedicated IFD and if the co-polarized signal is always on the primary receive card.

Live Angle Input - 0:None, 1:Sim, 2:TAGS, 3:S/D : 2

This setting is used to configure the input of live angles. In most situations, the angles will be coming in from the RCP via TAGS. The S/D option provides direct conversion of 3-wire synchro waveforms for the AZ and EL position angles. You may directly hookup AZ/EL synchros to the 12-pin input connector on the IO-62 standard backpanel when you choose S/D.

Primary RVP8/Rx PCI Card (-1:None) : 0

This setting configures which RVP8/Rx card will be used by the RVP8.

Primary RVP8/Tx PCI Card (-1:None) : -1

This setting configures which RVP8/Tx card will be used by the RVP8.

Primary I/O-62 PCI Card (-1:None) : 0

Run I/O-62 external line powerup tests:NO

This setting configures which RVP8/Rx card will be used by the RVP8. The I/O-62 external line powerup tests are used for debugging the backpanel and should be turned off during normal operation. When enabled, the backpanel receives a spread of signals which could cause problems in an operational environment (i.e. firing of the transmitter).

Provide IRIS RPC network status server: NO

The default value is NO in order to reduce network security concerns. When enabled, it opens up a network port.

PWINFO command enabled: No

The “Pulsewidth Information” user interface command can be disabled, thus further protecting the radar against inappropriate combinations of pulsewidth and PRF. This is a more safe setting in general, and is even more important when DPRT triggers are being generated. It can also be useful when running user code that is not yet fully debugged.

TRIGWF command enabled: NO

The “Trigger Waveform” user interface command can be disabled if you want to prevent the host computer from overwriting the RVP8’s stored trigger specifications. This is the default setting, based on the assumption that the built-in plotting commands would be used to configure the triggers. Answering “YES” will allow new waveforms to be loaded from the host computer.

RVP7 Emulation: No

The RVP8 implements a reasonably precise emulation of the RVP7 command set. This mode is useful because it allows an RVP8 to be plugged directly into a software system that used to run with an RVP7. All of the configuration steps that are new and unique to the RVP8 can be handled by the local TTY and Scope setups, thus making no demands on the user's system code for support. Answer this question "YES" for maximum compatibility with old driver software. However, if you are running IRIS version 6.11 or higher, then answer "NO" to enable using new RVP8 features as they are developed.

The RVP8 returns a version number of 25 when the processor is running in RVP7 compatibility mode. This fudged value will appear in the SCSI Inquiry Command reply, and in the GPARM parameter packet. Elsewhere, the correct RVP8 ROM version number will always appear. The reason for doing this is so that the RVP8 appears (to the host computer) to be a modern RVP7 with all of the latest opcodes and features.

3.2.2 Mp — Processing Options

Window- 0:User, 1:Rect, 2:Hamming, 3:Blackman : 0

Whenever power spectra are computed by the RVP8, the time series data are multiplied by a (real) window prior to computation of the Fourier Transform. You may use whichever window has been selected via SOPRM word #10, or force a particular window to be used.

R2 Processing- 0:Never, 1:User, 2:Always : 1

Controls R0/R1 versus R0/R1/R2 processing. Selecting "0" unconditionally disables the R2 algorithms, regardless of what the host computer requests in the SOPRM command. Likewise, selecting "2" unconditionally enables R2 processing. These choices allow the RVP8 to run one way or the other without having to rewrite the user code. This is useful for compatibility with existing applications.

Clutter Microsuppression- 0:Never, 1:User, 2:Always : 1

Controls whether individual "cluttery" bins are rejected prior to being averaged in range. Same interpretation of cases as for "R2 Processing" above.

2D Final Speckle/Unfold - 0:Never, 1:User, 2:Always : 1

The Doppler parameter modes (PPP, DFT, etc) include an optional 3x3 interpolation and speckle removal filter that is applied to the final output rays. This 2-dimensional filter examines three adjacent range bins from three successive rays in order to assign a value to the center point. Thus, for each output point, its eight neighboring bins in range and time are available to the filter. Only the *dBZ*, *dbT*, *Vel*, and *Width* data are candidates for this filtering step; all other parameters are processed using the normal 1-dimensional (three bins in range) speckle remover. See Section 5.3.3 for more details.

Unfold Velocity (Vh-Vl) - 0:Never, 1:User, 2:Always : 0

This question allows you to choose whether the RVP8 will unfold velocities using a simple ($V_{high} - V_{low}$) algorithm, rather than the standard algorithm described in Section 5.6. Bit-11 of SOPPRM word #10 is the host computer's interface to this function when the "1:User" case is selected (See Section 6.3).



Note: This setup question is included for research customers only. The standard unfolding algorithm should still be used in all operational systems because of its lower variance. For this reason, the factory default value of this parameter is "0:Never".

Process w/ custom trigs - 0:Never, 1:User, 2:Always : 0

This question allows you to choose whether the RVP8 will attempt to run its standard processing algorithms even when a custom trigger pattern has been selected via the SETPWF command. Generally it does not make sense to do this, so the default setting is "0:Never". Bit-12 of OPPRM word #10 is the host computer's interface to this function when the "1:User" case is selected (See Section 6.3).

Use High-SNR 16-bit packed timeseries format: Yes

This parameter provides an additional 6dB of SNR. It can be disabled to provide compatibility with legacy systems.

Minimum freerunning ray holdoff: 100% of dwell

This parameter controls the rate at which the RVP8 processes free-running rays in the DFT, DPRT, and Random Phase modes. This prevents rays from being produced at the full CPU limit or I/O limit of the processor (whichever was slower); which could result in highly overlapping data being output at an unusably fast rate. Note that this behavior will only occur when one of these non-PPP modes is chosen, and is then allowed to run without angle syncing. Such is likely the case for IRIS manual scans or during Passive IRIS mode.

To make these free-running modes more useful, you may establish a minimum holdoff between successive rays, expressed as a percentage of the number of pulses contributing to each ray. Choosing 100% (the default) will produce rays whose input data do not overlap at all, i.e., whose rate will be exactly the PRF divided by the sample size. Choosing 0% will give the unregulated behavior in which no minimum overlap is enforced and rays may be produced very quickly.

Limits: 0 to 100%

Linearized saturation headroom: 4.0 dB

The RVP8 uses a statistical saturation algorithm that estimates the real signal power correctly even when the IF receiver is overdriven (i.e., for input power levels above +4dBm). The algorithm works quite well in extending the headroom above the top end of the A/D converter, although the accuracy decreases as the overdrive becomes more severe. This parameter allows you to place an upper bound on the maximum extrapolation that will ever be applied. Choosing 0dB will disable the algorithm entirely.

Limits: 0 to 5dB

Apply amplitude correction based on Burst/COHO: YES
Time constant of mean amplitude estimator: 70 pulses

The RVP8 can perform pulse-to-pulse amplitude correction of the digital (I,Q) data stream based on the amplitude of the Burst/COHO input. Please see Section 5.1.6 for a complete discussion of this feature.

Limits: 10 to 500 pulses

IFD built-in noise dither source: -57.0dBm

This question will only appear if the processor is attached to a Rev.D RVP8/IFD that includes an out-of-band noise generator to supply dither power for the A/D converters. The available power levels are { Off, -57dBm, -37dBm, -32dBm, -27dBm, -22dBm, -19dBm }. The closest available level to your typed-in value will be used. You can observe the band-limited noise easily in the **Pr** plot to confirm its amplitude and spectral properties.

For standard operation, we recommend running at -57dBm. The problem higher levels of dither level is that, for certain choices of (I,Q) FIR filter, the stopband of the filter may not give enough attenuation to preserve the RVP8/IFD's inherent noise level. For example, the factory default 1MHz bandwidth Hamming filter has a stopband attenuation near DC of approximately 43dB. You can see this graphically at the right edge of the **Ps** menu. The in-band contribution of dither power is therefore approximately $(-37\text{dBm}) - 43\text{dB} = -80\text{dBm}$, which exceeds the A/D converter's 1MHz bandwidth noise of -81.5dBm.

TAG bits to invert	AZ:0000	EL:0000
TAG scale factors	AZ:1.0000	EL:1.0000
TAG offsets (degrees)	AZ:0.00	EL:0.00

The incoming TAG input bits may be selectively inverted via each of the 16-bit words. The values are displayed in Hex. Setting a bit will cause the corresponding AZ (bits 0–15) or EL (bits 16–31) lines to be inverted. Note that the SOPRM command also specifies TAG bits to invert. Both specifications are XOR'ed together to yield the net inversion for each TAG line.

The overall operations are performed in the order listed. Incoming bits are first inverted according to the two 16-bit XOR masks. This yields an unsigned 16-bit integer value which is then multiplied by the signed scale factor. The result is interpreted as a 16-bit binary angle (in the low sixteen bits), to which the offset angle is finally added.

As an example, suppose that the elevation angle input to the RVP8 was in an awkward form such as unsigned integer tenths of degrees, i.e., 0x0000 for zero degrees, 0x000a for one degree, 0x0e06 for minus one degree, etc. If we apply a scale factor of $65536/3600 = 18.2044$ to these units, we will get 16-bit binary angles in the standard format. If we further suppose that the input angle rotated “backwards”, we could take care of this too using a multiplier of -18.2044.

Interference Filter - 0:None, Alg.1, Alg.2, Alg.3: 1
Threshold parameter C1: 10.00 dB
Threshold parameter C2: 12.00 dB

The RVP8 can optionally apply an interference filter to remove impulsive-type noise from the demodulated (I,Q) data stream. See Section 5.1.4 for a complete description of this family of algorithms.

Provide WSR88D legacy BATCH major mode: YES
Maximum range to unfold: 600.0 km
Low-PRF bins range averaged on each side: 2
Overlay power - Refl:5.0dB Vel:8.0dB Width:12.0db
LowSamps = (0.00000 x HiSamps) + 6.00 :
LowPRF = (0.00000 x HiPRF) + 250.00 :

1. This is actually a fully general implementation of a Lo/Hi Surveillance/Doppler PRF unfolding scheme that provides all of the legacy features as special cases.

The parameters are defined as follows:

- The maximum range to unfold is given in km. This allows you to set an upper bound on how many Doppler trips will be unfolded according to the echoes seen in the surveillance data.
- The surveillance data set uses very few pulses and therefore is somewhat noisy. You may choose the number of bins that will be range averaged from both sides of these bins to provide a lower variance power estimate. A value of zero means “No averaging”, a value of one would average three points total, etc.
- The unfolding algorithm flags obscured range bins according to three different power thresholds for reflectivity, velocity, and width, and outputs these bits in the DB_FLAGS data parameter. Each of these thresholds is specified in deciBels.
- The fundamental RVP8 operating parameters (PRF, Sample Size, etc) all apply to the high PRF portion of the BATCH trigger waveform. The low PRF rate and sample size are derived from these high values using a slope and offset. In the example shown above, the slopes are both zero, so that the surveillance data will be fixed at 6-pulses and 250-Hz. Making the slopes nonzero would cause the low-PRF parameters to vary automatically if desired.

These setup parameters are accessible through the DSP driver using the new entry points *dspw_batchSetup()* and *dspw_batchSetup()*. These use the custom opcode that is defined separately by each major mode, so you may find *customUserOpcode_batch()* to be a useful model for how to build such things.

Polarization Params - Filtered:YES NoiseCorrected:YES
PhiDP - Negate: NO , Offset:0.0 deg
KDP - Length: 5.00 km
T/Z/V/W computed from: H-Xmt:YES V-Xmt:YES
T/Z/V/W computed from: Co-Rcv:YES Cx-Rcv:NO

The first question decides whether all polarization parameters will be computed from filtered or unfiltered data, and whether noise correction will be applied to the power measurements.

The second and third questions define the sign and offset corrections for Φ_{DP} and the length scale for *KDP*.

The fourth and fifth questions control how the standard parameters (Total Reflectivity, Corrected Reflectivity, Velocity, and Width) are computed in a multiple polarization system. Answering *YES* to *H-Xmt* and/or *V-Xmt* means that data from those transmit polarizations should be used whenever there is more than one choice available. Thus, these selections only apply to the Alternating and Simultaneous transmit modes. Likewise, answering *YES* to *Co-Rcv* and/or *Cx-Rcv* means to use the received data from the co-channel or cross-channel. The receiver question will only appear when dual simultaneous receivers have been configured.

A typical installation might use *H-Xmt:YES*, *V-Xmt:YES*, *Co-Rcv:YES*, *Cx-Rcv:NO*. This will compute (T/Z/V/W) from the co-polarized receiver using both H&V transmissions. Including both transmissions will decrease the variance of (T/Z/V/W); although some researchers prefer excluding *V-Xmt* because that is more standard in the literature. Also, if your polarizations are such that the main power is returned on the cross channel, then you will probably want *Co-Rcv:NO* and *Cx-Rcv:YES*.

DualRx - Sum H+V Time Series: NO

In dual-receiver systems, you may choose whether the (H+V) time series data consist of the sum of the “H” and “V” samples or the concatenation of half the “H” samples followed by half the “V” samples. The later is more useful when custom software is being used to analyze the data from the two separate receive channels.

3.2.3 Mf — Clutter Filters

Residual clutter LOG noise margin: 0.15 dB/dB

Whenever a clutter correction is applied to the reflectivity data, the LOG noise threshold needs to be increased slightly in order to continue to provide reliable qualification of the corrected values. The reason for this is that the uncertainty in the corrected reflectivity becomes greater after the clutter is subtracted away.

For example, if we observe 20dB of total power above receiver noise, and then apply a clutter correction of 19dB, we are left with an apparent weather signal power of +1dB above noise. However, the uncertainty of this +1dB residual signal is much greater than that of a pure weather target at the same +1dB signal level.

The “Residual Clutter LOG Noise Margin” allows you to increase the LOG noise threshold in response to increasing clutter power. In the previous example, and with the default setting of 0.15dB/dB, the LOG threshold would be increased by $19 \times 0.15 = 2.85\text{dB}$. This helps eliminate noisy speckles from the corrected reflectivity data.

Spectral Clutter Filters

```
-----
Filter #1 - Type:0(Fixed)      Width:1   EdgePts:2
Filter #2 - Type:0(Fixed)      Width:2   EdgePts:2
Filter #3 - Type:0(Fixed)      Width:3   EdgePts:3
Filter #4 - Type:1(Variable)   Width:3   EdgePts:2   Hunt:3
Filter #5 - Type:2(Variable)   (Gaussian Model) Spectrum width: 0.200 m/sec
Filter #6 - Type:2(Variable)   (Gaussian Model) Spectrum width: 0.300 m/sec
Filter #7 - Type:2(Variable)   (Gaussian Model) Spectrum width: 0.500 m/sec
```

These questions define the heuristic clutter filters that operate on power spectra during the DFT-type major modes. Filter #0 is reserved as “all pass”, and cannot be re-defined here. For filters #1 through #7, enter a digit to choose the filter type, followed by however many parameters that type requires. The three filter types are all described in detail in Section 5.2.4.

Fixed Width Filters (Type 0)

These are defined by two parameters. The “Width” sets the number of spectral points that are removed around the zero velocity term. A width of one will remove just the DC term; a width of two will remove the DC term plus one point on either side; three will remove DC plus two points on either side, etc. Spectral points are removed by replacing them with a linear interpolating line. The endpoints of this line are determined by taking the minimum of “EdgeMinPts” past the removed interval on each side.

Variable Width, Single Slope (Type 1)

The RVP8 supports variable-width frequency-domain clutter filters. These filters perform the same spectral interpolation as the fixed-width filters, except that their notch width automatically adapts to the clutter. The filters are characterized by the same *Width* and *EdgePts* parameters in the **Mf** menu, except that the *Width* is now interpreted as a minimum width. An additional parameter *Hunt* allows you to choose how far to extend the notch beyond *Width* in order to capture all of the clutter power. Setting *Hunt*=0 effectively converts a variable-width filter back into a fixed-width filter.

The algorithm for extending the notch width is based on the slope of adjacent spectral points. Beginning (*Width*-1) points away from zero, the filter is extended in each direction as long as the power continues to decrease in that direction, up to adding a maximum of *Hunt* additional points. If you have been running with a fixed *Width*=3 filter, you might try experimenting with a variable *Width*=2 and *Hunt*=1 filter. Perhaps the original fixed width was actually failing at times, but you were reluctant to increase it just to cover those rare cases. In that case, try selecting a variable *Width*=2 and *Hunt*=2 filter as an alternative. In general, make your variable filters “wider” by increasing *Hunt* rather than increasing *Width*. This will preserve more flexibility in how they can adapt to whatever clutter is present.

Gaussian Model Adaptive Processing (GMAP) (Type 2)

This type of processing is the most advanced form of clutter filtering and moment estimation (see Section 5.2.4). For GMAP processing, the only thing that needs be specified is the spectrum width of clutter. Note that the algorithm is not too sensitive to the exact value of this. Several widths should be configured to cover the antenna

rotation rates that are commonly used. It is useful to turn off clutter filtering (select the all pass filter #0) and then look at actual measurements of the clutter width while the antenna is rotating, e.g., using the ascope utility or application software such as the SIGMET IRIS system.

Whitening Parameters

Noise threshold for replacing a point: 1.20
Replacement value multiplier: 0.5000
SNR in tails, for determining width: 0.25

These questions control the adaptive whitening filter that is used by the Random Phase processing algorithms. A spectral point will be whitened if the ratio of its power to the noise power exceeds the "Noise threshold for replacing a point." The whitened point will consist of a complex value whose ARG is identical to that of the original point, and whose MAG is the product of the noise level with the "Replacement value multiplier" term. The nominal spectral width of the whitened region is a function of the power and width of the coherent signal, and the noise level. Assuming a Gaussian model, the "SNR in tails..." value is the ratio of the coherent power in the tails of the distribution to the noise level.

RPhase SQI Threshold Slope:0.50 Offset:-0.05

The two values in this question define a secondary SQI threshold that is used to qualify the LOG data during Random Phase processing. The secondary SQI level is computed by multiplying the primary user-supplied SQI threshold by the SLOPE, and adding the OFFSET. See also Section 5.7.3.

Limits: SLOPE: 0.0 to 2.0, OFFSET -2.0 to 1.0

3.2.4 Mt — General Trigger Setups

These questions are accessed by typing "Mt" with no additional arguments. They configure general properties of the RVP8 trigger generator

Pulse Repetition Frequency: 500.00 Hz

This is the Pulse Repetition Frequency of the internal trigger generator.

Limits: 50 to 20000Hz.

Transmit pulse width: 0

Limits: 0 to 3

Use external pretrigger: NO

PreTrigger active on rising edge: YES

PreTrigger is synchronous with IFD AQ clock: No

PreTrigger fires the transmitter directly: NO

When an external pretrigger is applied to the TRIGIN input of the RVP8, either the rising or falling edge of that signal initiates operation. This decision also affects which signal edge becomes the reference point for the pretrigger delay times given in the "Mt<n>" section.

Answer the second sub-question according to whether the radar transmitter is directly fired by the external pretrigger, rather than by one of the RVP8's trigger outputs. In other words, answer "YES" if the transmitter would continue running fine even if the RVP8 TRIGIN signal were removed. This information is used by the "L" and "R" subcommands of the "Pb" plotting command, i.e., when slewing left and right to find the burst pulse, the pretrigger delay will be affected rather than the start times of the six output triggers.

Number of user-defined output triggers: 6

This setting defines the number of user-defined output triggers.

Limit: 12 (including polarization output controls)

Number of polarization output controls: 2

This setting defines the number of polarization output controls.

2-way (Tx+Rx) total waveguide length: 0 meters

Use this question to compensate for the offset in range that is due to the length of waveguide connecting the transmitter, antenna, and receiver. You should specify the total 2-way length of waveguide, i.e., the span from transmitter to antenna, plus the span from antenna to receiver. The RVP8 range selection will compensate for the additional waveguide length to within plus-or-minus half a bin, and works properly at all range resolutions.

POLAR1 is high for vertical polarization : NO

POLAR2 is high for vertical polarization : NO

These questions define the logical sense of the two polarization control signals POLAR1 and POLAR2. In a dual-polarization radar POLAR1 should be used to select one of two possible states (nominally horizontal and vertical, but any other polarization pair may also be used). The control signal will either remain at a fixed level, or will alternate from pulse to pulse with a selectable transition point (See Section 3.2.5). POLAR2 is identical to POLAR1, but may be configured with a different polarity and switch point. This second signal could be used if the radar's polarization switch required more than one control line transition when changing states.

Quantize trigger PRT to $((1 \times AQ) + 0)$ clocks

It is possible to control the exact quantization of the PRT of the internal trigger generator. Normally the trigger PRT is chosen as the closest multiple of AQ (the acquisition clock period) that approximates the requested period. This question allows the possible PRT's to be constrained to $((N \times AQ) + M)$ clock cycles. This feature can be useful for synchronous receiver systems in which the trigger period must be some exact multiple of the COHO period.

Blank output triggers within AZ and EL sectors: NO

Sector #1	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #2	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #3	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #4	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #5	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #6	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #7	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0
Sector #8	- InUse:NO	AZ: 0.0,0.0	EL: 0.0,0.0

These settings can be modified to reduce erroneous transmissions into physical obstructions.

Blank output triggers via softplane sTrigBlank : NO

Blank triggers 1:YES 2:YES 3:YES 4:YES 5:YES 6:YES

These questions control trigger blanking based on the TAG0 input line. You first select whether the trigger blanking feature is enabled; and then optionally choose the polarity of TAG0 that will result in blanking, and which subset of the six user definable triggers are to be blanked.

Blank output triggers during noise measurement : NO

The RVP8 can inhibit the subset of blankable trigger lines whenever a noise measurement is taken. This will be forced whenever trigger blanking (based on TAG0) is enabled, but it can also be selected in general via this question. Since noise triggers must be blanked whenever trigger blanking is enabled, this question only appears if trigger blanking is disabled.

This question permits the state of the triggers during noise measurements to be consistent and known, regardless of whether the antenna happens to be within a blanked sector; and you have the additional flexibility of choosing blanked noise triggers all the time.

Rx-Fixed Triggers: #1:N #2:N #3:N #4:N #5:N #6:N P0:N P1:N Z:N

You have explicit control over which RVP8 trigger outputs are timed relative to the transmitter pre-fire sequence, versus those which are relative to the actual received target ranges. Triggers in the first category will be moved left/right by the "L/R" keys in the **Pb** plot, and will also be slewed in response to Burst Pulse Tracking. Triggers in the second category remain fixed relative to "receiver range zero", and are not affected by the "L/R" keys or by tracking.

This question specifies which triggers are Tx-relative and which are Rx-relative. Answer with a sequence of "Y" or "N" responses for each of the six trigger lines, for the two polarization control lines, and for the timing of the phase control lines. You should answer *No* for any trigger that is involved with the pre-fire timing of the transmitter. If you enable the Burst Pulse Tracker (Section 5.1.3) you will probably want to assign a *Yes* to some of your triggers so that they remain fixed relative to the burst itself.

It is very helpful to have these two categories of trigger start times. Triggers that fire the transmitter, either directly or indirectly, should all be moved as a group when hunting for the burst pulse and moving it to the center of the FIR window. However,

triggers that function as range strobes should be fixed relative to range zero, i.e., the center of that window, and the center of the burst. This distinction becomes important when the transmitter's pre-fire delay drifts with time and temperature.

Merge triggers to create composite waveforms: YES

```
Merge Trigger #1 into : #1:  #2:  #3:  #4:  #5:  #6:
Merge Trigger #2 into : #1:N #2:  #3:  #4:  #5:  #6:
Merge Trigger #3 into : #1:Y #2:  #3:  #4:  #5:  #6:
Merge Trigger #4 into : #1:  #2:Y #3:  #4:  #5:  #6:
Merge Trigger #5 into : #1:  #2:Y #3:  #4:  #5:  #6:
Merge Trigger #6 into : #1:  #2:  #3:  #4:  #5:  #6:
```

These questions allow you to merge the six user triggers together; resulting in trigger patterns that can be much more complex. In this example, Trigger #3 will be merged into Trigger #1; Trigger #3 will be unaltered, and Trigger #1 will be the "OR" of itself with Trigger #3. Likewise, Triggers #4 and #5 will be merged into Trigger #2 so that the later will contain three distinct pulses within each PRT. Answer each question with a sequence of up to six "Y" or "N" responses in order to set the merged destinations for each trigger line.

Note that the six triggers are still defined in the usual way in the **Mt<n>** menu, i.e., start time, width, etc. The only change is that you may now combine these individual pulse definitions into a more complex composite output waveform.

3.2.5 Mt<n> — Triggers for Pulswidth #n

These questions are accessed by typing "Mt", with an additional argument giving the pulswidth number. They configure specific trigger, transmit waveform, and FIR filter properties for the indicated pulswidth only.

```
Trigger #1 - Start:      0.00 usec
              #1 - Width:    1.00 usec      High:YES
Trigger #2 - Start:      0.00 usec + ( 0.500000 * PRT )
              #2 - Width:   10.00 usec      High:YES
Trigger #3 - Start:     -3.00 usec
              #3 - Width:    1.00 usec      High:YES
Trigger #4 - Start:     -2.00 usec
              #4 - Width:    1.00 usec      High:YES
Trigger #5 - Start:     -1.00 usec
              #5 - Width:    1.00 usec      High:YES
Trigger #6 - Start:     -5.00 usec + (-0.001000 * PRT )
              #6 - Width:    2.00 usec      High:NO
```

These parameters list the starting times (in microseconds relative to range zero), the widths (in microseconds), and the active sense of each of the six triggers generated by the internal trigger generator. Setting a width to zero inhibits the trigger on that line.

The Start Time can include an additional term consisting of the pulse period times a fractional multiplier between -1.0 and +1.0. This allows you to produce trigger patterns that would not otherwise be possible, e.g., a trigger that occurs half way

between every pair of transmitted pulses, and remains correctly positioned regardless of changes in the PRF. Enter this multiplier as "0" if you do not wish to use this term, and it will be omitted entirely from the printout..

In the above example, Trigger #2 is a 10.0 μ sec active-high pulse whose leading edge occurs precisely halfway between the zero-range of every pair of pulses. Likewise, Trigger #6 is a 2.0 μ sec active-low pulse whose falling edge is nominally 5.0 μ sec prior to range zero, but which is advanced by 1.0 μ sec for every millisecond of trigger period. All other triggers behave normally, and have fixed starting times that do not vary with trigger period.

Some subtleties of these variable start times are:

- The PRT multipliers can only be used in conjunction with the RVP8's internal trigger generator. The PRT-relative start times are completely disabled whenever an external trigger source is chosen from the **Mt** menu.
- When PRT-relative triggers are plotted by the **Pb** command, the active portion of the trigger will be drawn cross-hatched and at a location computed according to the current PRF. The cross-hatching serves as a reminder that the actual location of that trigger may vary from it's presently plotted position.
- The PRT multiplier for a given pulse is applied to the interval of time between that pulse and the next one. This distinction is important whenever the RVP8 is generating multiple-PRT triggers, e.g., during DPRT mode, or during Dual-PRF processing. Multipliers from 0.0 to +1.0 are generally safe to use because they shift the trigger into the same pulse period that originally defined it. For example, a start time of $(0.0 \mu\text{sec} + (0.98 * \text{PRT}))$ would position a trigger 98% of the way up to the next range zero. But, if -0.98 were used, and if the period of the previous pulse was shorter than the current one, then that shorter period would become incorrect (longer) as a result of having to fit in the very early trigger.

A small but important detail is built into the algorithm for producing the six user trigger waveforms. It applies whenever a) the trigger period is internally determined, i.e., the external pretrigger input is not being used, and b) the overall span of the six trigger definitions combined does not fit into that period. What happens in this case is that any waveforms that do not fit will be zeroed (not output) so that the desired period is preserved. This means that you can define triggers with large positive start times, and they will pop into existence only when the PRF is low enough to accommodate them.

For example, if Trigger #2 is defined as a 200.0 μ sec pulse starting at +400.0 μ sec, then that trigger would be suppressed if the PRF were 2000Hz, but it would be present at a PRF of 1000Hz. Whenever a trigger does not completely fit within the overall period it is suppressed entirely. Thus, even though the +400.0 μ sec start time is still valid at 2000Hz, the entire 200.0 μ sec pulse would not fit, and so the pulse is eliminated altogether.

Start limits: -5000 to $5000 \mu\text{sec}$. Width limits: 0 to $5000 \mu\text{sec}$.

Maximum number of Pulses/Sec: 2000.0

Maximum instantaneous 'PRF' : 2000.0 (/Sec)

These are the PRF protection limits for this pulsewidth.

The wording of the "Maximum number of Pulses/Sec" question serves as a reminder that the number shown is not only an upper bound on the PRF, but also a duty cycle limit when DPRT mode is enabled.

The "Maximum instantaneous 'PRF'" question allows you to configure the maximum instantaneous rate at which triggers are allowed to occur, i.e., the reciprocal of the minimum time between any two adjacent triggers. This parameter is included so that you can limit the maximum DPRT trigger rate individually for each pulsewidth. Note that the maximum instantaneous PRF can not be set lower than the maximum number of pulses per second.

PRF limits: 50 to 20000Hz.

External pretrigger delay to range zero: 3.00 usec

Range Zero is time at which the signal from a target at zero range would appear at the radar receiver outputs. This parameter adjusts the delay from the active edge of the external trigger to range zero. It is important that this delay be correct when the RVP8 is operating with an external trigger, since the zero range point is a fixed time offset from that trigger. When the transmitter is driven from the internal trigger signals, those signals themselves are adjusted (see Burst Pulse alignment procedures) to accomplish the alignment of range zero.

Limits: 0.1 to 1000 μ sec.

Range mask spacing: 125.00 meters

The range resolution of the RVP8 is determined by the decimation factor of the digital matched FIR filter that computes "I" and "Q". This decimation factor is the ratio of the filter's input and output data rates, i.e., the output rate is some integer divisor of the IFD Acquisition Clock (See **Mc** Section). For the legacy RVP7 IFD operating at its standard frequency of 35.9751MHz, the available range resolutions (in meters) are: 25.0, 28.3, 36.7, 50.0, 58.3, 66.7, 75.0, 83.3, 91.7, 100.0, 108.3, 116.7, 125.0, and 133.3. The RVP8 IFD operating in the 72MHz range provides twice the resolution between steps, and four times the number of steps.

The ranges that are selected by the bit mask in the LRMSK command are spaced according to the range resolution that is chosen here. Also, the upper limit on the impulse response length of the matched FIR filter (see below) is constrained by the range resolution. If you choose a range resolution that can not be computed at the present filter length, then a message of the form: "Warning: Impulse response shortened from 72 to 42 taps" will appear.

Limits: 25 to 1000 meters.

FIR-Filter impulse response length: 1.33 usec

The RVP8 computes "I" and "Q" using a digital FIR (Finite Impulse Response) matched filter. The length of that filter (in microseconds) is chosen here.

The filter length should be based on several considerations:

- It should be at least as long as the transmitted pulsewidth. If it were shorter, then some of the returned energy would be thrown away when “I” and “Q” are computed at each bin. The SNR would be reduced as a result.
- It should be at least as long as the range bin spacing. The goal here is to choose the longest filter that retains statistical independence among successive bins. If the filter length is less than the bin spacing, then no IF samples would be shared among successive bins, and those bins would certainly not be correlated.
- It should be “slightly longer” than either of the above bounds would imply, so that the filter can do a better job of rejecting out-of-band noise and spurious signals. The SNR of weak signals will be improved by doing this.

In practice, a small degree of bin-to-bin correlation is acceptable in exchange for the filter improvements that become possible with a longer impulse response. The FIR coefficients taper off to zero on each end; hence, the power contributed by overlapping edge samples is minimal. SIGMET recommends beginning with an impulse response length of 1.2–1.5 times the pulsewidth or bin spacing, whichever is greater.

The maximum possible filter length is bounded according to the range resolution that has been chosen; a finer bin spacing leaves less time for computing a long filter. For the RVP8 Rev.A processor, the filter length must be less than 2.92 μ sec at 125-meter resolution; for Rev.B and higher this limit increases to 6.67 μ sec.

NOTE: Cascade filter software is being contemplated that will extend the maximum impulse response length to at least 50 μ sec. This is of interest when very long (uncoded CW) transmitted pulses are used.

Burst Freq Estimator- Length: 1.33 usec, Start: 0.00 usec

This estimator is mostly used with the Pb (plotting commands) and can be referenced in Section 4.3.2.

FIR-Filter prototype passband width: 0.503 MHz

This is the passband width of the ideal lowpass filter that is used to design the matched FIR bandpass filter. The actual bandwidth of the final FIR filter will depend on 1) the filter's impulse response length, and 2) the design window used in the process. The actual 3dB bandwidth will be:

- Larger than the ideal bandwidth if that bandwidth is narrow and the FIR length is too short to realize that degree of frequency discrimination. In these cases it may be reasonable to increase the filter length.
- Smaller than the ideal bandwidth if the FIR length easily resolves the frequency band. This is because of the interaction within the filter's transition band of the ideal filter and the particular design window being used. For example, for a Hamming window and sufficiently long filter length, the ideal bandwidth is an approximation of the 6dB (not 3dB) attenuation point. Hence, the 3dB width is narrower than the ideal prototype width.

This parameter should be tuned using the TTY output and interactive visual plot from the “Ps” command. The actual 3dB bandwidth is shown there, so that it can be compared with the ideal prototype bandwidth.

Limits: 0.05 to 10.0 MHz.

Output control 4-bit pattern: 0001

These are the hardware control bits for this pulsewidth. The bits are the 4-bit binary pattern that is output on PWBW0:3

Bit Limits: 0 to 15 (input must be typed in decimal)

Current noise level: -75.00 dBm

Powerup noise level: -75.00 dBm

-or-

Current noise levels - PriRx: -75.00 dBm, SecRx: -75.00 dBm

Powerup noise levels - PriRx: -75.00 dBm, SecRx: -75.00 dBm

These questions allow you to set the current value and the power-up value of the receiver noise level for either a single or dual receiver system. The noise level(s) are shown in dBm, and you may alter either one from the TTY. The power-up level(s) are assigned by default when the RVP8 first starts up, and whenever the RESET opcode is issued with Bit #8 set. Likewise, the current noise level is revised whenever the SNOISE opcode is issued. These setup questions are intended for applications in which the RVP8 must operate with a reasonable default value, up until the time that an SNOISE command is actually received. They may also be used to compare the receiver noise levels during normal operation, which serves as a check that each FIR filter is behaving as expected when presented with thermal noise.

Transmitter phase switch point: -1.00 usec

This is the transition time of the RVP8's phase control output lines during random phase processing modes. The switch point should be selected so that there is adequate settling time prior to the burst/COHO phase measurement on each pulse. This question only appears if the PHOUT[0:7] lines are actually configured for phase control (See Section 3.2.1).

Limits: -500 to 500 μ sec.

Polarization switch point for POLAR1: -1.00 usec

Polarization switch point for POLAR2: 1.00 usec

The RVP8's POLAR1 and POLAR2 digital output lines control the polarization switch in a dual-polarization radar. During data processing modes in which the polarization alternates from pulse to pulse, the transition points of these control signals are set by these two questions. The values are in microseconds relative to range zero; the same units used to define the start times of the six user triggers. The logical sense of POLAR1 and POLAR2 is set by questions described in Section 3.2.4.

Limits: -500 to 500 μ sec.

3.2.5.1 Special Options for Tx Synthesis

Several of the dialogs described in the previous section will be modified when the RVP8 is equipped with an RVP8/Tx Digital Transmitter Card that has been configured for Tx waveform synthesis in the **Mz** menu. In this case, each of the RVP8's four "pulsewidths" can select an entirely different type of transmit waveform and associated matched receiver.

For example, PW-0 and PW-1 could transmit conventional 0.5 μ sec and 2.0 μ sec CW pulses that are received using the bandpass filters described in Section 3.2.5. But within this same system, PW-2 and PW-3 could be further configured as, perhaps, 20 μ sec and 40 μ sec compressed non-linear FM waveforms. This makes it very easy for application software such as **ascope** to transparently switch between radically different Tx waveforms simply by requesting a different "pulsewidth" for each one.

The following questions will appear in the **Mt<n>** menu (immediately after the *Range Mask Spacing* question) when digital Tx waveforms are being synthesized.

Tx Waveform - 0:CWPulse, 1:LinFM, 2:NLFM : 2

The RVP8 supports three standard Tx waveforms: a conventional fixed-frequency CW pulse, a linear FM chirp, and non-linear FM. The CWPulse can be used as a pulsed Doppler waveform in all the same ways that a Klystron or Magnetron system having a traditional pulse forming network would be used. The linear and non-linear FM waveforms, however, are compressed pulses that are intended to be transmitted by a wide-bandwidth Klystron/TWT/SolidState amplifier.

Note: The RVP8 internal APIs permit code developers to create arbitrary waveforms for transmission. The three types mentioned above are the out-of-the-box selections that are standard on all RVP8 processors.

Bandwidth of transmit pulse: 3.25 MHz

Pulselength of transmit pulse: 15.00 usec

These questions select the bandwidth and pulse length of the Tx waveform. The bandwidth value represents the true spectrum width of the complete waveform, i.e., including all the effects of whatever frequency modulation and amplitude modulation the waveform happens to use. Thus, a spectrum analyzer (or the RVP8's **Ps** plot) would show an overall spectrum width equal to this desired value.

Likewise, the pulse length value represents the entire time duration of the waveform, including whatever amplitude modulations may be included at the tails.

Zero offset of transmit pulse: 0.00 usec

The Tx waveform is normally synthesized with its center lined up with range zero. If the radar's high-power amplifier had zero delay, this would serve to define the middle of the transmit pulse as range zero, which is the usual RVP8 convention. This offset question is provided so that the RVP8/Tx output waveform can be shifted in time to compensate for whatever delays are present in the radar's IF/RF electronics.

Note: This transmit pulse timing offset is typically checked via the **Pb** plot by making sure that the Tx burst is centered within the FIR data window.

TxWave MIN tuning params: 0.0000, 0.0000, 0.0000

TxWave MAX tuning params: 1.0000, 1.0000, 1.0000

TxWave tuning parameters: 0.9500, 1.0000, 0.0490

The RVP8 uses three real-valued tuning parameters to make the synthesis of complex waveforms more flexible. Each waveform class can be altered and fine tuned with up to three degrees of freedom, making it possible for a single class (e.g., the non-linear FM class) to generate a huge variety of actual waveforms. These adjustable constants also form the basis of the automatic waveform optimization procedure described for the **Pa** command in Section 4.6.2.

Each of the three parameters has a minimum value, a maximum value, and a current value, all of which can be changed from this menu. The Min/Max limits are used within the **Pa** command to maintain sensible bounds as the parameters are adjusted. In general, the Min/Max values will be entered from the **Mt<n>** menu, but the actual values will be tuned using either manual or automatic procedures found in the **Pa** command.

The CWPulse class of waveforms do not use any of the tuning parameters because the Tx waveform is completely determined by the desired bandwidth and pulsewidth, i.e., there are no remaining degrees of freedom to adjust. Thus, these three questions do not appear in the CWPulse case.

The linear FM class is also entirely specified by just the bandwidth and pulsewidth values, and does not reference any of the tuning parameters. However, the non-linear FM class is the most flexible of all, and references all three tuning parameters as follows:

- Parameters #1 and #2 are the (X,Y) location of the non-linear “breakpoint” for the FM curve. Referring to the white plot line in Figure 4–9, the Time/Frequency behavior of the pulse can be drawn in a coordinate system whose abscissa ranges from –1 to +1 over the complete time duration of the pulse, and whose ordinate ranges from –1 to +1 over the complete frequency span of the pulse.

The class of non-linear FM curves always pass through the points (–1,–1), (0,0), and (1,1), i.e., they begin at the lowest frequency at the start of the pulse, end at the highest frequency when the pulse completes, and pass through the origin (to maintain symmetry across both halves of the pulse). Between the points (0,0) and (1,1) the curves also pass through the tunable (X,Y) “breakpoint” defined by the first two parameters. In other words, the positive-time portion of the FM curve consists of two linear segments; one from (0,0) to (X,Y), and the other from (X,Y) to (1,1). By tuning the breakpoint we create a diverse class of FM modulations, but all of them adhere to the physical bandwidth and pulsewidth limits imposed by the earlier setup questions. Note that to maintain symmetry, the breakpoint is also mirrored on the negative-time side as line segments from (–1,–1) to (–X,–Y), and from (–X,–Y) to (0,0).

- Parameter #3 specifies the X location of the start of the amplitude taper of the non-linear FM waveform. For example, setting X to 0.95 will result in a pulse having full amplitude over the middle 95% of its duration, but then having raised cosine amplitude weighting applied to the leading and trailing 5% of its edges.

Some examples may be helpful:

P1 = 0.0, P2 = 0.0, P3 = 1.0

P1 and P2 place the FM breakpoint at the origin. But the FM curve passes through that point anyway, so the response reverts to linear FM. P3 indicates that amplitude modulation should not be applied until the very end of the pulse, and thus will not occur at all. The resulting waveform is therefore linear FM having abrupt On/Off transitions.

P1 = 0.9, P2 = 0.7, P3 = 1.0

During the middle 90% of the waveform's duration the frequency chirp uses 70% of its available bandwidth. Then, within the 10% pulse tails, the remaining 30% of the bandwidth suddenly gets covered. No amplitude modulation is applied. Pulses of this type have been studied theoretically, but do not perform very well for a given total bandwidth that includes the leading/trailing "ears".

P1 = 0.9, P2 = 1.0, P3 = 0.8

The entire frequency band is chirped within the middle 90% of the pulse duration, so that the frequency remains constant in the 10% pulse tails. An amplitude modulation is also applied over 20% of the pulse tails, i.e., encompassing both the ends of the chirp and the entire constant frequency intervals. Pulses of this type have superior sidelobe behavior and fit very neatly within their prescribed bandwidth limits. We recommend using non-linear FM waveforms that combine chirp limits and amplitude modulation in this manner.

3.2.6 Mb — Burst Pulse and AFC

These questions are accessed by typing “Mb”. They set the parameters that influence the phase and frequency analysis of the burst pulse, and the operation of the AFC feedback loop.

Receiver Intermediate Frequency: 30.0000 MHz

This is the center frequency of the IF receiver and burst pulse waveform.

- With the legacy RVP7 IFD, the RVP8 can operate at an intermediate frequency within any of the alias bands 22–32MHz, 40–50MHz, and 58–68MHz.
- With the CAT-5E IFD, the RVP8 can operate at an intermediate frequency within any of the alias bands 8–32MHz and 40–68MHz

These bands are delineated by 4MHz safety zones on either side of integer multiples of half the IFD's sampling frequency. The value entered here implicitly defines the band being used.

Limits: 6 to 72 MHz.

Primary Receiver Intermediate Frequency: 30.0000 MHz

Secondary Receiver Intermediate Frequency: 24.0000 MHz

These alternate questions will replace the previous question whenever the RVP8's dual-receiver mode is selected. You should enter the two intermediate frequencies for your primary and secondary (nominally horizontal and vertical polarized) receivers. Note that you can easily swap receiver channels merely by exchanging the two frequency values.

IF increases for an approaching target: YES

The intermediate frequency is derived at the receiver's front end by a microwave mixer and sideband filter. The filter passes either the lower sideband or the upper sideband, and rejects the other. Depending on which sideband is chosen, an increase in microwave frequency may either increase (STALO below transmitter) or decrease (STALO above transmitter) the receiver's intermediate frequency. This question influences the sign of the Doppler velocities that are computed by the RVP8.

PhaseLock to the burst pulse: YES

This question controls whether the RVP8 locks the phase of its synthesized “I” and “Q” data to the measured phase of the burst pulse. For an operational magnetron system this should always be “YES”, since the transmitter's random phase must be known in order to recover Doppler data. The “NO” option is appropriate for non phase modulated Klystron systems in which the RVP8/IFD sampling clock is locked to the COHO. It is also useful for bench testing in general. In these “NO” cases the phase of “I” and “Q” is determined relative to the stable internal sampling clock in the RVP8/IFD module.

Minimum power for valid burst pulse: -15.0 dBm

This is the minimum mean power that must be present in the burst pulse for it to be considered valid, i.e., suitable for input into the algorithms for frequency estimation and AFC. The reporting of burst pulse power is described in Section 4.3; the value entered here should be, perhaps, 8 dB less. This insures that burst pulses will still be properly detected even if the transmitter power fades slightly.

The mean power level of the burst is computed within the narrowed set of samples that are used for AFC frequency estimation. The narrow subwindow will contain only the active portion of the burst, and thus a mean power measurement is meaningful. The full FIR window would include the leading and trailing pulse edges and would not produce a meaningful average power. Since radar peak power tends to be independent of pulse width, this single threshold value can be applied for all pulsewidths.

Limits: -60 to +10 dBm.

Design/Analysis Window- 0:Rect, 1:Hamming, 2:Blackman : 1

You may choose the window that is used in 1) the design of the FIR matched filter, and 2) the presentation of the power spectra for the various scope plots. Choices are rectangular, Hamming, and Blackman; the Hamming window being the best overall choice. The Blackman window is useful if you are trying to see plotted spectral components that are more than 40dB below the strongest signal present. It is especially useful in the "Pr" plot when a long span of data are available. FIR filters designed with the Blackman window will have greater stopband attenuation than those designed with the Hamming window, but the wider main lobe may be undesirable. The rectangular window is included mostly as a teaching tool, and should never be used in an operational setting.

Settling time (to 1%) of burst frequency estimator: 5.0 sec

The burst frequency estimator uses a 4th order correlation model to estimate the center frequency of the transmitted pulses. Each burst pulse will typically occupy approximately one microsecond; yet the frequency estimate feeding the AFC loop needs to be accurate to, perhaps, 10KHz. Obviously this accuracy can not be achieved using just one pulse. However, several hundred of the (unbiased) individual estimates can be averaged to produce an accurate mean. This averaging is done with an exponential filter whose time constant is chosen here.

Limits: 0.1 to 120 seconds.

Lock IFD sampling clock to external reference: NO

This question determines the usage of the shared SMA connector that is labeled "AFC/(CLK)" on the RVP8/IFD. It is generally *not* necessary to phase lock the IFD sampling clock to the radar system clock, since very good stability is obtained from the burst phase measurements during normal operation. However, two cases that benefit from clock locking are 1) using the RVP8 in a klystron system where an external trigger is provided, and 2) dual-receiver systems in which computation of Φ_{DP} is important.

The following two questions will appear only if you have requested that the IFD sampling clock be locked to an external clock reference. See Section 2.2.12 for a description of the hardware setups that must accompany this selection.

PLL ratio of (1/1) ==> Input reference at 17.9876 MHz

The VCXO phase-locked-loop (PLL) in the RVP8/IFD can work with any input reference clock whose frequency is a rational multiple (P/Q) of half the desired sampling frequency, i.e., the center frequency of the VCXO that is entered in the “mc” command (Section 3.2.1). This question allows this ratio to be established. In general, the best PLL performance will be attained when the ratio is reduced to lowest terms, e.g., use a ratio of 6/5 rather than 12/10. For example, assume:

VCXO center frequency $f_{\text{samp}} = 36 \text{ MHz}$

Reference clock frequency $f_{\text{ref}} = 10 \text{ MHz}$

$$f_{\text{ref}} = \frac{P}{Q} \frac{f_{\text{samp}}}{2} \quad ; \quad \frac{P}{Q} = \frac{2 f_{\text{ref}}}{f_{\text{samp}}} = \frac{2 \cdot 10 \text{ MHz}}{36 \text{ MHz}} = \frac{5}{9}$$

In this case enter the numbers “5 9”. The proper input reference frequency (e.g., 10.0000 MHz) should be displayed after you enter the values. The “v” command can be used to verify that the PLL is “Okay”.

Limits: 1 to 128 for both numerator and denominator.

VCXO has positive frequency deviation: YES

Most VCXOs have positive frequency deviation, i.e., their output frequency increases with increasing input control voltage. This question will generally be answered “yes”, but is included to accommodate the other case as well. The PLL will not lock, and will be completely unstable, if the wrong choice is made.

Enable AFC and MFC functions: YES

AFC is required in a magnetron system to maintain the fixed intermediate frequency difference between the transmitter and the STALO. AFC is not required in a klystron system since the transmitted pulse is inherently at the correct frequency.

The following rather long list of questions will appear only if AFC and MFC functions have been enabled.

AFC Servo- 0:DC Coupled, 1:Motor/Integrator : 0

The AFC servo loop can be configured to operate with an external Motor/Integrator frequency controller, rather than the usual direct-coupled FM control. This type of servo loop is required for tuned magnetron systems in which the tuning actuator is moved back and forth by a motor, but remains fixed in place when motor drive is removed. These systems require that the AFC output voltage (motor drive) be zero when the loop is locked; and that the voltage be proportional to frequency error while tracking. Please see Section 3.2.6.1 for more details.

Wait time before applying AFC: 10.0 sec

After a magnetron transmitter is first turned on, it may be several seconds or even minutes until its output frequency becomes stable. It would not make sense for the AFC loop to be running during this time since there is nothing gained by chasing the startup transient. This question allows you to set a holdoff delay from the time that valid burst pulses are detected to the time that the AFC loop actually begins running.

Limits: 0 to 300 seconds.

AFC hysteresis -- Inner: 5.0 KHz, Outer: 15.0 KHz

These are the frequency error tolerances for the AFC loop. The loop will apply active feedback whenever the outer frequency limit is exceeded, but will hold a fixed level once the inner limit has been achieved. The hysteresis zone minimizes the amount of thrashing done by the feedback loop. The AFC control voltage will remain constant most of the time; making small and brief adjustments only occasionally as the need arises.

AFC outer tolerance during data processing: 50.0 KHz

In general, the AFC feedback loop is active only when the RVP8 is not processing data rays. This is because the Doppler phase measurements are seriously degraded whenever the AFC control voltage makes a change. To avoid this, the AFC loop is only allowed to run in between intervals of sustained data processing. This is fine as long as the host computer allows a few seconds of idle time every few minutes; but if the RVP8 were constantly busy, the AFC loop would never have a chance to run. This question allows you to place an upper bound on the frequency error that is tolerated during sustained data processing. AFC is guaranteed to be applied whenever this limit is exceeded.

Limits: 15 to 4000 KHz.

AFC feedback slope: 0.0100 D-Units/sec / KHz
AFC minimum slew rate: 0.0000 D-Units/sec
AFC maximum slew rate: 0.5000 D-Units/sec

These questions control the actual feedback computations of the AFC loop.

The overall span of the AFC output voltage is set by Gain and Offset potentiometers on the RVP8/IFD module (See Section 2.2.11). The control level that is applied to the AFC's 16-bit Digital-to-Analog converter is specified here in "D-Units", i.e., arbitrary units ranging from -100 to +100 corresponding to the complete span of the D/A converter. Since the D-Unit corresponds in a natural way to a percentage scale, the shorter "%" symbol is sometimes used.

AFC feedback will be applied in proportion to the frequency error that the algorithm is attempting to correct. The feedback slope determines the sensitivity and time constant of the loop by establishing the AFC's rate of change in (D-Units / sec) per thousand Hertz of frequency error. For example, a slope of 0.01 and a frequency error of 30KHz would result in a control voltage slew of 0.3 D-Units per second. At that rate it would take approximately 67 seconds for the output voltage to slew one tenth of its total span ($20 \text{ D-Units} / (0.3 \text{ D-Units} / \text{sec}) = 67 \text{ sec}$). AFC is intended to track very slow drifts in the radar system, so response times of this magnitude are reasonable.

Keep in mind that the feedback slew is based on a frequency error which itself is derived from a time averaging process (see Burst Frequency Estimator Settling Time described above). The AFC loop will become unstable if a large feedback slope is used together with a long settling time constant, due to the phase lag introduced by the averaging process. Keep the loop stable by choosing a small enough slope that the loop easily comes to a stop within the inner hysteresis zone.

See Section 3.2.6.1 for more information about these slope and slew rate parameters.

AFC span- [-100%,+100%] maps into [-32768 , 32767]
AFC format- 0:Bin, 1:BCD, 2:8B4D: 0, ActLow: NO
AFC uplink protocol- 0:Off, 1:Normal, 2:PinMap : 1

The RVP8's implementation of AFC has been generalized so that there is no difference between configuring an analog loop and a digital loop. The AFC feedback loop parameters are setup the same way in each case; the only difference being the model for how the AFC information is made available to the outside world. Many types of interfaces and protocols thus become possible according to how these three questions are answered. AFC output follows these three steps:

- The internal feedback loop uses a conceptual [-100%,+100%] range of values. However, this range may be mapped into an arbitrary numeric span for eventual output. For example, choosing the span from -32768 to +32767 would result in 16-bit AFC, and 0 to 999 might be appropriate for 3-digit BCD; but any other span could also be selected from the full 32-bit integer range.
- Next, an encoding format is chosen for the specified numeric span. The result of the encoding step is another 32-bit pattern which represents the above numeric value. SIGMET will make an effort to include in the list of supported formats all custom encodings that our customers encounter from their vendors.

Available formats include straight binary, BCD, and mixed-radix formats that might be required by a specialized piece of equipment. The "8B4D" format encodes the low four decimal digits as four BCD digits, and the remaining upper bits in binary. For example, 659999 base-10 would encode into 0x00419999 Hex.

- Finally, an output protocol is selected for the bit pattern that was produced by encoding the numeric value. The bits may be written to the eight RVP8 backpanel RS232 outputs, or sent on the uplink as a value to be received by the RVP8/IFD and converted to an analog voltage. Yet another option is for the bits to be sent on the uplink and received by the DAFC, which supports arbitrary remapping of its output pins.

To summarize: the internal AFC feedback level is first mapped into an arbitrary numeric span, then encoded using a choice of formats, and finally mapped into an arbitrary set of pins for digital output. We are hopeful that this degree of flexibility will allow easy hookup to virtually any STALO synthesizer that one might encounter.

PinMap Table (Type '31' for GND, '30' for +5)

```

-----
Pin01:00   Pin02:01   Pin03:02   Pin04:03   Pin05:04
Pin06:05   Pin07:06   Pin08:07   Pin09:08   Pin10:09
Pin11:10   Pin12:11   Pin13:12   Pin14:13   Pin15:14
Pin16:15   Pin17:16   Pin18:17   Pin19:18   Pin20:19
Pin21:20   Pin22:21   Pin23:22   Pin24:23   Pin25:24
FAULT status pin (0:None): 0, ActLow: NO

```

These questions only appear when the “PinMap” uplink protocol has been selected. The table assigns a bit from the encoded numeric word to each of the 25 pins of the RVP8/DAFC module. For example, the default table shown above simply assigns the low 25 bits of the encoded bit pattern to pins 1-25 in that order. You may also pull a pin high or low by assigning it to +5 or GND. Note that such assignments produce a logic-high or logic-low signal level, not an actual power or ground connection. The latter must be done with actual physical wires.

One of the RVP8/DAFC pins can optionally be selected as a Fault Status indicator. You may choose which pin to use for this purpose, as well as the polarity of the incoming signal level. Note that the standard RVP8/DAFC module only supports the selection of pins 1, 3, 4, 13, 14, and 25 as inputs. This setup question allows you to choose any pin, however, because it does not know what kind of hardware may be listening on the uplink and what its constraints might be.

Burst frequency increases with increasing AFC voltage: NO

If the frequency of the transmit burst increases when the AFC control voltage increases, then answer this question “Yes”; otherwise answer “No”. When this question is answered correctly, a numerical increase in the AFC drive (D-Units) will result in an increase in the estimated burst frequency. If the AFC loop is completely unstable, try reversing this parameter.

Mirror AFC voltage on- 0:None, 1:I, 2:Q : 0

AFC/MFC can be mirrored on a backpanel output of the main chassis using this question. When either “I” or “Q” is selected, the AFC/MFC voltage will be present on the corresponding BNC output, and the other output will be used for scope plotting. This configuration would be useful, for example, in a dual-receiver magnetron system that needs a phase locked acquisition clock in the RVP8/IFD, but also needs an AFC tuning voltage to control the transmit frequency. When “None” is selected, scope plotting will revert to its normal “Q” output.

The voltage range of the “I” and “Q” outputs is approximately ± 1 Volt, and is not adjustable. When AFC/MFC is mirrored on these lines, you will probably need to add an external Op-Amp circuit to adjust the voltage span and offset to match your RF components. We also recommend that you add significant low-pass filtering (cutoff at 3Hz) to remove any power line noise or crosstalk that may be originating within the RVP8 chassis.

Enable Burst Pulse Tracking: YES

This question enables the Burst Pulse Tracking algorithm that is described in Section 5.1.3. Remarkably, for such an intricate new feature, there are no additional parameters to configure. The characteristic settling times for the burst are already defined elsewhere in this menu, and the tracking algorithm uses dynamic thresholds to control the feedback.

Enable Time/Freq hunt for missing burst: No

Number of frequency intervals to search: 5

Settling time for each frequency hop: 0.25 sec

Automatically hunt immediately after being reset: YES

Repeat auto hunt every: 60.00 sec

These questions configure the process of hunting for a missing burst pulse. The trigger timing interval that is checked during Hunt Mode is always the maximum $\pm 20\mu\text{sec}$; hence no further setup questions are needed to define the hunting process in time. The hunt in frequency is a different matter. The overall frequency range will always be the full -100% to $+100\%$ AFC span; but the number of subintervals to check must be specified, along with the STALO settling time after making each AFC change. With the default values shown, AFC levels of -66% , -33% , 0% , $+33\%$, and $+66\%$ will be tried, with a one-quarter second wait time before checking for a valid burst at each AFC setting.

You should choose the number of AFC intervals so that the hunt procedure can deduce an initial AFC level that is within a few megaHertz of the correct value. The normal AFC loop will then take over from there to keep the radar in tune. For example, if your radar drifts considerably in frequency so that the AFC range had to be as large as 35MHz, then choosing fifteen subintervals might be a good choice. The hunt procedure would then be able to get within 2.3MHz of the correct AFC level. The settling time can usually be fairly short, unless you have a STALO that wobbles for a while after making a frequency change. Note that hunting in frequency is not allowed for Motor/Integrator AFC loops, and the two AFC questions will be suppressed in that case.

The RVP8 can optionally begin hunting for a missing burst pulse immediately after being reset, but before any activity has been detected from the host computer. This might be useful in systems that both drift a lot and generally have their transmitter *On*. However, this option is really included just as a work around; the correct way for a burst pulse hunt to occur is via an explicit request from the host computer which “knows” when the pulse really should be present. Blindly hunting in the absence of that knowledge can not be done because there are many reasons why the burst pulse may legitimately be missing, e.g., during a radar calibration.

The automatic hunt for the burst pulse will always run at least once whenever the feature is enabled. The automatic hunting ceases, however, as soon as any activity is detected from the host computer. Only use this feature on radars with a serious drift problem in their burst pulse timing.

Simulate burst pulse samples: NO

The RVP8 can simulate a one microsecond envelope of burst samples. This is useful only as a testing and teaching aid, and should never be used in an operational system.

A two-tone simulation will be produced when the RVP8 is setup in dual-receiver mode. The pulse will be the sum of two transmit pulses at the primary and secondary intermediate frequencies. To make the simulation more realistic, the two signal strengths are unequal; the primary pulse is 3dB stronger than the secondary pulse.

Frequency span of simulated burst: 27.00 MHz to 32.00 MHz

The simulated burst responds to AFC just as a real radar would. The frequency span from minimum AFC to maximum AFC is given here.

3.2.6.1 AFC Motor/Integrator Option

The question "*AFC Servo- 0:DC Coupled, 1:Motor/Integrator*" selects whether the AFC loop runs in the normal manner (direct control over frequency), or with an external Motor/Integrator type of actuator. The question "*AFC minimum slew request:...*" provides additional control when interfacing to mechanical actuators whose starting and sustaining friction needs to be overcome.

The DC-Coupled AFC loop questions (changes shown in bold) are:

```
AFC Servo- 0:DC Coupled, 1:Motor/Integrator : 0
Wait time before applying AFC: 10.0 sec
AFC hysteresis- Inner: 5.0 KHz, Outer:15.0 KHz
AFC outer tolerance during data processing: 50.0 KHz
AFC feedback slope:      0.0100 D-Units/sec / KHz
AFC minimum slew rate:  0.0000 D-Units/sec
AFC maximum slew rate:  0.5000 D-Units/sec
```

and the Motor/Integrator loop questions are:

```
AFC Servo- 0:DC Coupled, 1:Motor/Integrator : 1
Wait time before applying AFC: 10.0 sec
AFC hysteresis- Inner: 5.0 KHz, Outer: 15.0 KHz
AFC outer tolerance during data processing: 50.0 KHz
AFC feedback slope:      1.0000 D-Units / KHz
AFC minimum slew request: 15.0000 D-Units
AFC maximum slew request: 90.0000 D-Units
```

Notice that the physical units for the feedback slope and slew rate limits are different in the two cases. In the DC-Coupled case the AFC output voltage controls the frequency directly, so the units for the feedback and slew parameters use *D-Units/Second*. In the Motor/Integrator case, the AFC output determines the rate of change of frequency; hence *D-Units* are used directly.

The above example illustrates typical values that might be used with a Motor/Integrator servo loop. The feedback slope of 1.0 *D-Units/KHz* means that a frequency error of 100KHz would produce the full-scale (100 *D-Units*) AFC output. But this is modified by the minimum and maximum slew requests as follows:

- A zero *D-Unit* output will always be produced whenever AFC is locked.
- When AFC is tracking, the output drive will always be at least ± 15 *D-Units*. This minimum non-zero drive should be set to the sustaining drive level of the motor actuator, i.e., the minimum drive that actually keeps the motor turning.
- When AFC is tracking, the output drive will never exceed ± 90 *D-Units*. This parameter can be used to limit the maximum motor speed, even when the frequency error is very large.

The AFC Motor/Integrator feedback loop works properly even if the motor has become stuck in a “cold start”, i.e., after the radar has been turned off for a period of time. The mechanical starting friction can sometimes be larger than normal, and additional motor drive is required to break out of the stuck condition. But once the motor begins to turn at all, then the normal AFC parameters (minimum slew, maximum slew, feedback slope) all resume working properly. The algorithm operates as follows:

- Whenever AFC correction is being applied, the RVP8 calculates how long it would take to reach the desired IF frequency at the present rate of change. For example, if we are 1MHz away from the desired IF frequency, and the measured rate of change of the IF burst frequency is 20KHz/sec, then it will be 50 seconds until the loop reaches equilibrium.
- Whenever the AFC loop is in Track-Mode, but the time to equilibrium is greater than two minutes, then the “Minimum Slew” parameter will be slowly increased. The idea is to gradually increase the starting motor drive whenever it appears that the IF frequency is not actually converging toward the correct value, i.e., the motor is stuck.
- As soon as the frequency is observed to begin changing, such that the desired IF would be reached in less than two minutes, then the “Minimum Slew” parameter is immediately put back to its correct setup value. The loop then continues to run properly using its normal setup values.

Manual Frequency Control (MFC) operates unchanged in both of the AFC servo modes. Whenever MFC is enabled in the **Ps** command, it always has the effect of directly controlling the output voltage of the AFC D/A converter. The MFC mode can be useful when testing the motor response under different drive levels, and when determining the correct value for the minimum slew request.

3.2.7 M+ — Debug Options

A collection of debugging options has been added to the RVP8 to help users with the development and debugging of their applications code. For the most part, these options should remain disabled during normal radar operation. These questions are included so that the RVP8 can be placed into unusual, and perhaps occasionally useful, operating states.

Noise level for simulated data: -50.0 dB

This is the noise level that is assumed when simulated “I” and “Q” data are injected into the RVP8 via the **LSIMUL** command. The noise level is measured relative to the power of a full-scale complex (I,Q) sinusoid, and matches the levels shown on the slide pots of the ASCOPE digital signal simulator.

Limits: -100dB to 0dB

Simulate output rays: NO

Answering “YES” to this question causes the RVP8 to output bands of simulated data. The bands can occupy a selectable range interval, and span a selectable interval of data values.

Start bin:0, Width:10 bins, Bands:16

This question is only asked if we are simulating output rays. The Start Bin chooses the bin number (origin zero) where the simulated bands will begin. The width of each band (in bins), and the total number of bands are also selected. The upper limit for all parameters is the maximum bin count for the RVP8.

Limits: Start: 0-Max, Width: 1-Max, Bands: 1-Max

Start data value:0, Increment:16

This question is only asked if we are simulating output rays. The data value that will be assigned to the first simulated band, and the data increment from one band to the next, are selected. The permissible values are from 0 to 65535, i.e., the full unsigned 16-bit integer range. This full range is useful when simulating 16-bit output data; for the more typical 8-bit output formats, only the low byte of the start and increment are significant.

Limits: 0 to 65535

3.2.8 Mz — Transmissions and Modulations

These questions are used to configure the 8-Bit phase modulation codes that may be used to control the phase of a coherent transmitter. The RVP8/Tx will output a pseudo-random sequence of phase codes that are chosen from a specified set of available codes, i.e., all 8-bit patterns that are valid for the phase modulation hardware. The random sequence is output only when the RVP8 is in one of its random phase processing modes (time series or parameter). At all other times, a fixed “idle” phase code pattern is output. See also Sections 3.2.1 and 3.2.5 where related phase control questions are found.

8-Bit code to output when idle: 0x00

This is the bit pattern to be output whenever the RVP8 is not in a random phase processing mode. Note that this “idle” code does not have to be one of the “active” codes that are enabled below.

Selection of Valid 8-Bit States

00-0F:	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10-1F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20-2F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30-3F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40-4F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50-5F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60-6F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
70-7F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80-8F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
90-9F:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A0-AF:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B0-BF:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C0-CF:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D0-DF:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

E0-EF: - - - - -
F0-FF: - - - - -

This set of questions defines the subset of active 8-bit codes that are valid states for the transmit phase modulator. Answer each line with a sequence of Y's or N's to indicate whether the corresponding 8-bit code is enabled. Only the codes that appear with a "Y" will be used by the RVP8; the "-" indicates an unused code. The "-" character was used instead of "N" so that the visual contrast of the printed table would be improved.

As an example, if your klystron transmitter has an octant phase modulator that is controlled by three digital lines, you might enable phase codes zero through seven, and then cable the modulator to the low three bits of the 8-bit code. The upper five bits would not need to be used in this case.

3.3 Advanced Options

3.3.1 “*” — Sample current noise levels

The “*” samples current noise levels from the receiver and then subtracts that noise from subsequent measurements. More information is provided in the Sample Noise Level (SNOISE) section 6–6.

3.3.2 “@” — Display/Change current Major Mode

This command provides developers with a simple way of switching modes enabling on-the-fly testing of code. The top level RVP8 operating modes are described in the documentation of the SOPRM command word #9. This question allows you to use the mode that has been selected by that command, or to force the use of a particular mode.

3.3.3 “~” — Burst-In / IF-In Swap Command (Rev.D IFD)

The “~” command swaps the Burst and IF inputs at the IFD. Requests to toggle the state are made from the top level as follows:

```
RVP8> ~  
IFD Burst/IF Inputs are: SWAPPED  
RVP8> ~  
IFD Burst/IF Inputs are: NORMAL
```

The selection remains in effect for the duration of the setup session, but then returns to NORMAL upon exiting the TTY monitor. The “~” command is very handy because it allows the **Pb**, **Pr**, and **Ps** plotting commands to easily run with one input or the other. Here are two examples of how this might be useful.

- When checking the range alignment on a Klystron system, the **Pb** plot can not be used in the usual way to center the Tx burst because a continuous-wave COHO (rather than a burst pulse) is typically used as the phase reference in these systems. However, if you swap the Burst and IF inputs, you can then use the **Pb** command to view and center the received leakage of the Tx pulse, and thus locate range zero.
- When setting up the AFC loop, you can use your RF signal generator to simulate the transmitter's frequency, and then run the loop with swapped RVP8/IFD inputs. The AFC servo will then hunt and follow the siggen frequency supplied via the receiver. You can then make step changes in that frequency to verify that the loop responds properly.

Note that the same input swapping function is also available via the RVP8/IFD toggle switches. However, those switches may be located far away from the operator's terminal; hence, the command interface is still a valuable addition. The “~” command can only be used with the new Rev.D RVP8/IFD; the command is unimplemented, and will not even show up in the “Help” list, when earlier receivers are connected.